

FORM PTO-1390 (Modified)  
(REV 11-98)

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

ATTORNEY'S DOCKET NUMBER

**TRANSMITTAL LETTER TO THE UNITED STATES  
DESIGNATED/ELECTED OFFICE (DO/EO/US)  
CONCERNING A FILING UNDER 35 U.S.C. 371**

**L9289.01130**

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR  
New Application **09/807625**

INTERNATIONAL APPLICATION NO.  
**PCT/JP00/05351**

INTERNATIONAL FILING DATE  
**August 10, 2000**

PRIORITY DATE CLAIMED  
**August 18, 1999**

TITLE OF INVENTION

**SOFTWARE REWRITING METHOD AND SOFTWARE REWRITING APPARATUS**

APPLICANT(S) FOR DO/EO/US

**Isao ISOGAI**

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

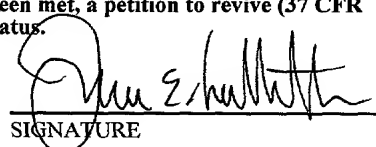
1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☐ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
  - a. ☐ is transmitted herewith (required only if not transmitted by the International Bureau).
  - b. ☒ has been transmitted by the International Bureau.
  - c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☒ A copy of the International Search Report (PCT/ISA/210).
8. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
  - a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
  - b. ☐ have been transmitted by the International Bureau.
  - c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
  - d. ☐ have not been made and will not be made.
9. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
10. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
11. ☐ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

**Items 13 to 20 below concern document(s) or information included:**

13. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☒ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☐ A **FIRST** preliminary amendment.
16. ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
17. ☐ A substitute specification.
18. ☐ A change of power of attorney and/or address letter.
19. ☐ Certificate of Mailing by Express Mail
20. ☒ Other items or information:

**Claim for Priority with PCT/IB/304  
PCT/IB/308  
PCT/RO/101**

6 APR 2001

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR 1.53(a)(2)) New Application <b>09/807625</b>		INTERNATIONAL APPLICATION NO. <b>PCT/JP00/05351</b>		ATTORNEY'S DOCKET NUMBER <b>L9289.01130</b>																					
21. The following fees are submitted: <b>BASIC NATIONAL FEE ( 37 CFR 1.492 (a) (1) - (5) ) :</b> <input type="checkbox"/> Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO ..... <b>\$1,000.00</b> <input checked="" type="checkbox"/> International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO ..... <b>\$860.00</b> <input type="checkbox"/> International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO ..... <b>\$710.00</b> <input type="checkbox"/> International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) ..... <b>\$690.00</b> <input type="checkbox"/> International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) ..... <b>\$100.00</b> <b>ENTER APPROPRIATE BASIC FEE AMOUNT =</b>				<b>CALCULATIONS PTO USE ONLY</b>																					
Surcharge of <b>\$130.00</b> for furnishing the oath or declaration later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).				<b>\$860.00</b>																					
<table><tr><td>CLAIMS</td><td>NUMBER FILED</td><td>NUMBER EXTRA</td><td>RATE</td></tr><tr><td>Total claims</td><td>10 - 20 =</td><td>0</td><td>x \$18.00</td></tr><tr><td>Independent claims</td><td>3 - 3 =</td><td>0</td><td>x \$80.00</td></tr><tr><td colspan="3">Multiple Dependent Claims (check if applicable).</td><td><input type="checkbox"/></td></tr><tr><td colspan="3"><b>TOTAL OF ABOVE CALCULATIONS</b></td><td><b>=</b></td></tr></table>				CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE	Total claims	10 - 20 =	0	x \$18.00	Independent claims	3 - 3 =	0	x \$80.00	Multiple Dependent Claims (check if applicable).			<input type="checkbox"/>	<b>TOTAL OF ABOVE CALCULATIONS</b>			<b>=</b>	<b>\$0.00</b>	
CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE																						
Total claims	10 - 20 =	0	x \$18.00																						
Independent claims	3 - 3 =	0	x \$80.00																						
Multiple Dependent Claims (check if applicable).			<input type="checkbox"/>																						
<b>TOTAL OF ABOVE CALCULATIONS</b>			<b>=</b>																						
Reduction of 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28) (check if applicable). <input type="checkbox"/>				<b>\$0.00</b>																					
<b>SUBTOTAL</b>				<b>= \$860.00</b>																					
Processing fee of <b>\$130.00</b> for furnishing the English translation later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 CFR 1.492 (f)).				<b>\$0.00</b>																					
<b>TOTAL NATIONAL FEE</b>				<b>= \$860.00</b>																					
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) (check if applicable). <input checked="" type="checkbox"/>				<b>\$40.00</b>																					
<b>TOTAL FEES ENCLOSED</b>				<b>= \$900.00</b>																					
				<b>Amount to be: refunded \$</b>																					
				<b>charged \$</b>																					
<input checked="" type="checkbox"/> A check in the amount of <b>\$900.00</b> to cover the above fees is enclosed.																									
<input type="checkbox"/> Please charge my Deposit Account No. _____ in the amount of _____ to cover the above fees. A duplicate copy of this sheet is enclosed.																									
<input checked="" type="checkbox"/> The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. <b>19-4375</b> A duplicate copy of this sheet is enclosed.																									
<b>NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.</b>																									
<b>SEND ALL CORRESPONDENCE TO:</b>																									
<div><div><b>James E. Ledbetter</b> <b>STEVENS, DAVIS, MILLER &amp; MOSHER, L.L.P.</b> <b>1615 L Street, NW, Suite 850</b> <b>Tel: 202/785-0100</b> <b>Fax: 202/408-5200</b></div><div> <b>SIGNATURE</b> <b>James E. Ledbetter</b> <b>NAME</b> <b>28,732</b> <b>REGISTRATION NUMBER</b> <b>April 16, 2001</b> <b>DATE</b></div></div>																									

## DESCRIPTION

SOFTWARE REWRITING METHOD AND SOFTWARE REWRITING  
APPARATUS

## 5 Technical Art

The present invention relates to a software rewriting method and a software rewriting apparatus.

## 10 Background Art

A conventional software rewriting method includes for example a method (1) for carrying out rewriting when software necessary for rewriting (hereinafter referred to as "old software") is not  
15 in execution and a method (2) for temporarily stopping execution forcibly during execution of old software.

In the method (1), for example, a terminal dedicated to software rewriting is connected to an  
20 apparatus on which old software operates and power source of the apparatus is once turned off and then turned on. At this time, when it is detected that new software to be rewritten (hereinafter referred to as "new software") is sent to the dedicated  
25 terminal, software for rewriting software provided in the apparatus stops the operation of old software and rewrites old software to new software in a rewriting mode different from a normal processing

mode. After that, the dedicated terminal is disconnected from the apparatus and power source of the apparatus is turned off/on, whereby new software is executed in the apparatus.

5 In the method (2), for example, in the normal processing a central processing unit (hereinafter referred to as "CPU") in the apparatus that old software operates temporarily stops execution of old software by an interrupt and the like and  
10 performs rewriting, and releases the stopping state after rewriting, thereby executing new software.

However, the conventional method (1) has a problem in which the apparatus cannot be used for the original purpose since the apparatus cannot  
15 carry out the normal processing during the rewriting. Also, the conventional method (2) has a problem in which the use of the apparatus for the original purpose is not disturbed but the normal processing of the apparatus is temporarily stopped.

20

#### Disclosure of Invention

It is an object of the present invention is to provide a software rewriting method and a software rewriting apparatus capable of rewriting software  
25 without stopping execution of the software during execution of the software.

The inventor of the present invention considered that the reason why rewriting is

performed when the software is not executed or when execution of the software is stopped lies in the point that the rewriting is executed in accordance with an execution state of the software. The  
5 inventor found out that the software can be divided into executed parts and unexecuted parts, and achieved the present invention.

In order to attain the above object, in the present invention, parts of the software are  
10 detected which are not executed, and the parts are rewritten sequentially.

#### Brief Description of Drawings

FIG. 1 is a general block diagram illustrating  
15 a schematic configuration of a software rewriting apparatus according to a first embodiment of the present invention;

FIG. 2 is a view explaining a software rewriting operation according to the first embodiment of the  
20 present invention; and

FIG. 3 is a view explaining a software rewriting operation according to a second embodiment of the present invention.

#### 25 Best Mode for Carrying Out the Invention

Embodiments of the present invention will be specifically explained with reference to the drawings attached hereto.

(First embodiment)

In the software rewriting method and the software rewriting apparatus according to the present embodiment, a plurality of processing  
5 blocks to be subjected to rewriting in software is divided into blocks in execution and unexecuted blocks, and the unexecuted blocks are rewritten.

The following will explain the software rewriting method and the software rewriting  
10 apparatus according to the first embodiment of the present invention. FIG. 1 is a general block diagram illustrating a schematic configuration of the software rewriting apparatus according to the first embodiment of the present invention.

15 In a CPU 101, a normal processing section 102, a rewrite processing section 103, and a control section 104 are operated.

The normal processing section 102 has an execution block storage 105. The normal processing  
20 section 102 extracts software necessary for processing that is currently performed from various kinds of software stored in a software storage 108, and expands it in the execution block storage 105 to execute normal processing. It is noted that  
25 normal processing which is herein referred indicates all processing except software rewrite processing performed in the CPU 101.

The rewrite processing section 103 has a

rewrite block storage 106 and a rewrite table 107. The rewrite processing section 103 updates old software stored in the software storage 108 by new software sent from a transfer section 109 in accordance with an instruction from the control section 104, thereby rewriting old software to new software.

The rewrite block storage 106 stores a new program sent from the transfer section 109 until rewrite processing is ended.

The rewrite table 107 is a table, which is used when the rewrite processing section 103 rewrites software, and a table for managing a software execution state. The configuration of the rewrite table 107 will be described later.

The control section 104 monitors a degree of load of processing to be executed by the CPU 101. The control section 104 instructs the rewrite processing section 103 to perform rewrite processing when an amount of processing executed at the normal processing section 102 is reduced and the load of processing to be executed by the CPU 101 becomes low, so that the CPU 101 can have a space where processing other than the normal processing can be carried out. The control section 104 also instructs the rewrite processing section 103 to stop rewrite processing when the amount of processing executed by the normal processing section 102 is

increased. Thus, since the software rewriting apparatus executes rewrite processing when the degree of load of processing executed by the CPU becomes low, it is possible to rewrite software without exerting an influence upon normal processing executed by the CPU.

The software storage 108 stores various kinds of software executed by the normal processing section 102.

10 The transfer section 109 converts new software transmitted from an external of the apparatus via cable or radio waves to data format that can be handled by the CPU 101, and transfers it to the rewrite block storage 106.

15 An explanation will be next given of an operation of the software rewriting apparatus having the above configuration with reference to FIG. 2. FIG. 2 is a view explaining a software rewriting operation according to the first embodiment of the present invention.

20 Regarding various kinds of software stored in the software storage 108, the overall software is divided to a plurality of processing blocks on a processing unit basis. Software is expanded in the execution block storage 105 or rewritten by the rewrite processing section 103 for each processing block unit.

Now, it is herein assumed that the overall old

software to be rewritten is divided to 10 blocks as shown in FIG. 2 (A).

Normally, in rewriting software, the overall software is not rewritten and only the block  
5 (hereinafter referred to as "old block"), which is necessary to be rewritten, is rewritten to improve a reduction in rewriting time.

A new block (hereinafter referred to as "new block") to be rewritten that is sent to the rewrite  
10 block storage 106 from the transfer section 109 is stored in the rewrite block storage 106 until the rewrite processing is ended.

The rewrite processing section 103 generates the rewrite table 107 as illustrated in FIG. 2(B)  
15 when detecting that the new blocks are sent to the rewrite block storage 106. The rewrite table is a table that has an item indicative of block numbers of new blocks and an item indicative of execution states as to whether or not the old blocks  
20 corresponding to the new blocks are expanded in the currently execution block storage 105 and they are in execution.

Now, it is herein assumed that blocks 1 to 3 as new blocks are transferred to the rewrite block  
25 storage 106. The item indicative of block numbers of new blocks is filled with blocks 1 to 3 as shown in FIG. 2(B). It is noted that the item indicative of execution state has not been filled yet at this

stage.

When the amount of processing executed at the normal processing section 102 is reduced and the load of processing executed by the CPU 101 becomes low, the rewrite processing section 103 is instructed to carry out rewrite processing by the control section 104.

Since the control section 104 monitors blocks being expanded in the execution block storage 105 to be executed, the control section 104 instructs the rewrite processing section 103 to carry out rewrite processing, and at the same time, notifies the rewrite processing section 103 of the block numbers, which have been expanded in the currently execution block storage 105 and which are in execution. Additionally, in the execution block storage 105, a plurality of blocks necessary for processing currently executed at the normal processing section 102 is combined and expanded.

Now, it is herein assumed that blocks, which have been expanded in the execution block storage 105 and which are in execution, are old blocks 1 to 3 as illustrated in FIG. 2(C)(1) in the old software of FIG. 2(A).

Next, the rewrite processing section 103 compares the block numbers notified from the control section 104 with block numbers stored in the rewrite table 107, and fills the item indicative of

execution state. Since the notified block numbers are 1 to 3 and the block numbers stored in the rewrite table 107 are also 1 to 3, the rewrite processing section 103 fills the item of all execution states of the rewrite table 107 with "in execution" as illustrated in FIG. 2 (B) (1).

Then, the rewrite processing section 103 extracts blocks, which are not currently executed, among from new blocks stored in the rewrite block storage 106 by referring to the rewrite table 107, and rewrites the old blocks stored in the software storage 108 to new blocks. Now, since all blocks 1 to 3 are in execution as illustrated in FIG. 2 (B) (1), the rewriting is not carried out.

After that, when the amount of processing executed at the normal processing section 102 is reduced again, the rewrite processing section 103 is instructed to carry out rewrite processing by the control section 104. At this time, it is assumed that blocks, which have been expanded in the execution block storage 105 and which are in execution, are blocks 8 to 10 as illustrated in FIG. 2(C)(2).

Next, the rewrite processing section 103 fills the item indicative of execution state by the same operation as mentioned above. Since the notified block numbers are 8 to 10 and the block numbers stored in the rewrite table are 1 to 3, the rewrite processing section 103 fills the item of all

execution states of the rewrite table 107 with "in stop" indicative of non-executing as illustrated in FIG. 2 (B) (2).

Then, the rewrite processing section 103  
5 extracts blocks 1 to 3, which are not currently executing, from the rewrite block storage 106 by referring to the rewrite table 107, and rewrites the old blocks 1 to 3 stored in the software storage 108 to new blocks 1 to 3.

10 After rewriting, the rewrite processing section 103 erases the blocks stored in the rewrite block storage 106 and the content of the rewrite table 107.

This embodiment shows the configuration in  
15 which the control section 104 always monitors the degree of load of CPU 101, and when the load of processing executed by the CPU 101 becomes low, instructs the rewrite processing section 103 to carry out rewrite processing. However, the  
20 following configuration may be provided. Namely, when new blocks are transferred to the rewrite block storage 106 and a rewrite request sent from the rewrite processing section 103 is transmitted to the control section 104, the control section 104 surveys  
25 the degree of load of CPU 101. Then, when the load of processing executed by the CPU 101 is low, the control section 104 may instruct the rewrite processing section 103 to carry out rewrite

processing.

Thus, according to the software rewriting method and the software rewriting apparatus according to this embodiment, the plurality of processing blocks to be rewritten in software is divided into blocks in execution and blocks in stop and rewrites the blocks in stop, so that the software can be rewritten during normal processing without stopping execution of the software.

10

(Second embodiment)

The difference between the software rewriting apparatus of this embodiment and that of the first embodiment is that the rewrite table has an item indicative of a rewriting state.

The following will explain the software rewriting method and the software rewriting apparatus according to the second embodiment with reference to FIG.1 and FIG. 3. FIG. 3 is a view explaining a software rewriting operation according to the second embodiment of the present invention.

Since the configuration and operation of the software rewriting apparatus of this embodiment are substantially the same as those of the first embodiment, the specific explanation of the same configuration and operation is omitted.

The rewrite processing section 103 generates the rewrite table 107 as illustrated in FIG. 3(B)

when detecting that a new block is sent to the  
rewrite block storage 106. This rewrite table is a  
table in which an item indicative of rewriting state  
is added to the rewrite table of the first  
5 embodiment.

Now, it is herein assumed that blocks 1 to 3  
and 8 to 10 as new blocks are transferred to the  
rewrite block storage 106. The item of block number  
of the rewrite table 107 is filled with blocks 1 to  
10 3 and 8 to 10 as shown in FIG. 3(B). At the time of  
filling blocks 1 to 3 and 8 to 10 in the rewrite table,  
all rewriting states are filled as "unfinished."

It is noted that the item of execution state  
has not been filled yet at this stage.

15 When the amount of processing executed at the  
normal processing section 102 is reduced and the  
load of processing to be executed by the CPU 101  
becomes low, the rewrite processing section 103 is  
instructed to carry out rewrite processing and is  
20 notified of block numbers, which have been expanded  
in the execution block storage 105 and which are in  
execution, by the control section 104.

Now, it is herein assumed that blocks, which  
have been expanded in the execution block storage  
25 105 and which are in execution, are blocks 1 to 3  
similar to the first embodiment as illustrated in  
FIG. 3(C) (1) in the old software of FIG. 3(A).

Next, the rewrite processing section 103

compares the block numbers notified from the control section 104 with the block numbers stored in the rewrite table 107, and fills the item indicative of execution state. The notified block numbers are 1 to 3 and the block numbers stored in the rewrite table 107 are 1 to 3 and 8 to 10. For this reason, the rewrite processing section 103 fills the item indicative of execution state of the rewrite table 107 with "in execution" relating to block numbers 1 to 3 and "in stop" relating to block numbers 8 to 10 as illustrated in FIG. 3(B)(1).

After that, the rewrite processing section 103 extracts blocks, which are not currently executing and which have not been finished rewriting, among new blocks stored in the rewrite block storage 106 from the rewrite block storage 106, and rewrites the old blocks stored in the software storage 108 to new blocks by referring to the rewrite table 107.

Now, it is herein assumed that rewriting of all new blocks is "unfinished", blocks 1 to 3 are "in execution", and blocks 8 to 10 are "in stop." Therefore, the rewrite processing section 103 extracts new blocks 8 to 10 from the rewrite block storage 106 and rewrites old blocks 8 to 10 stored in the software storage 108 to new blocks 8 to 10.

It is noted that the rewriting state of blocks 8 to 10 illustrated in FIG. 3(B)(1) remains "unfinished" until the rewrite processing section

103 finishes rewriting blocks 8 to 10.

Then, the rewrite processing section 103 changes the rewriting state of blocks 8 to 10 of the rewrite table 107 from "unfinished" to "finished" as illustrated in FIG. 3(B)(1) when the rewrite processing section 103 finishes rewriting blocks 8 to 10.

Next, when the amount of processing executed at the normal processing section 102 is increased, the control section 104 instructs the rewrite processing section 103 to temporarily stop rewrite processing. The rewrite processing section 103 temporarily stops rewrite processing in accordance with this instruction. At this time, the content of the rewriting state of the rewrite table 107 shows "unfinished" in connection with blocks 1 to 3 and "finished" in connection with blocks 8 to 10 as illustrated in FIG. 3 (B)(1). Therefore, when the rewrite processing section 103 receives the instruction of rewrite processing from the control section 104 again, the rewrite processing section 103 is able to judge that the rewrite processing section 103 does not have to rewrite blocks 8 to 10, which have been already rewritten, and rewrites only blocks 1 to 3, which have not been rewritten.

Then, when the amount of processing executed at the normal processing section 102 is reduced again, the rewrite processing section 103 is

instructed to carry out rewrite processing by the control section 104. At this time, it is assumed that blocks, which have been expanded in the execution block storage 105 and which are in execution, are  
5 blocks 8 to 10 as illustrated in FIG. 3(C) (2), similar to the first embodiment.

Next, the rewrite processing section 103 fills the item indicative of execution state with "in stop" relating to blocks 1 to 3 and "in execution" relating to blocks 8 to 10 as illustrated in FIG.  
10 3(B)(2) since the notified block numbers are 8 to 10.

After that, the rewrite processing section 103 extracts blocks 1 to 3, which are not currently  
15 executing and which have been "unfinished" rewriting, from the rewrite block storage 106 by referring to the rewrite table 107, and rewrites the old blocks 1 to 3 stored in the software storage 108 to new blocks 1 to 3. After rewriting the blocks 1  
20 to 3, the rewrite processing section 103 changes the rewriting state of blocks 1 to 3 of the rewrite table 107 from "unfinished" to "finished" as illustrated in FIG. 3(B)(2).

Then, the rewrite processing section 103  
25 erases the blocks stored in the rewrite block storage 106 and the content of the rewrite table 107 when confirming that all rewriting states of the rewrite table 107 are changed to "finished."

According to the software rewriting method and the software rewriting apparatus of this embodiment, since the rewrite table has the item indicative of the rewriting state, the plurality of processing blocks to be rewritten in software is divided into blocks finished rewriting and blocks unfinished rewriting, and only the blocks unfinished rewriting can be subjected to rewriting. This eliminates the need for carrying out the rewriting by one rewrite processing, and makes it possible to divide the rewriting to a plurality of times when there are a lot of blocks to be written. Moreover, regarding the blocks finished rewriting, since it is unnecessary to carry out the rewriting again, the load of rewrite processing can be reduced and the rewrite processing can be efficiently carried out.

The software rewriting apparatus according to the first and second embodiments can be applied to a communication terminal apparatus such as a mobile station apparatus in the radio communication system. In such an application, for example, if the communication terminal apparatus is the mobile station apparatus, software is transmitted to the mobile station apparatus from the base station apparatus via radio waves using a control channel on which a control signal is always transmitted. Then, the mobile station apparatus can carry out the rewriting of software at the time when the load of

CPU processing is relatively low, for example, a standby status. In the conventional case, when the rewriting of software is needed, the mobile station apparatus must be collected to carry out the rewriting of software or a user of the mobile station apparatus must visit the store where the terminal for rewriting is provided to carry out the rewriting. Therefore, the mobile station apparatus cannot be used in communications during the rewriting.

According to the present invention, the rewriting of software can be carried out in the normal use status of the mobile station apparatus without having the user of the mobile station apparatus recognize that the rewriting of software has been carried out.

The software rewriting apparatus of the first and second embodiments can be applied to a communication terminal apparatus connected via cable, a computer apparatus connected via cable, and a computer apparatus connected via radio waves.

As explained above, according to the present invention, software can be rewritten without stopping execution of software during execution of software.

This application is based on the Japanese Patent Application No. HEI 11-231468 filed on August 18, 1999, entire content of which is expressly incorporated by reference herein.

### Industrial Applicability

The present invention can be applied to a communication terminal apparatus such as a mobile station apparatus in a radio communication system. Also, the present invention can be applied to a communication terminal apparatus connected via cable, a computer apparatus connected via cable, and a computer apparatus connected via radio waves.

## CLAIMS

1. A software rewriting method that detects unexecuted parts of software to be rewritten during execution of the software to be rewritten and  
5 rewrites the unexecuted parts sequentially.
2. The software rewriting method according to claim 1, wherein the software to be rewritten is software having one block or software, which is divided into a plurality of blocks, and the respective blocks are  
10 classified into executing blocks and unexecuted blocks, and the unexecuted blocks are sequentially rewritten.
3. The software rewriting method according to claim 2, wherein rewriting blocks are temporarily stored  
15 in a memory, the rewriting blocks are compared with the executing blocks, and when blocks corresponding to the rewriting blocks are unexecuted, the corresponding blocks of the software to be rewritten are sequentially rewritten to the rewriting blocks.
- 20 4. The software rewriting method according to claim 3, wherein it is determined as to whether or not the corresponding blocks of the software to be rewritten are finished rewriting, and no rewriting of the corresponding blocks, which have been rewritten,  
25 are carried out again.
5. A software rewriting apparatus comprising:  
a software storage for storing software having one block or a plurality of divided blocks;

a processor for expanding the blocks to be executed;

a block storage for temporarily storing rewriting blocks;

5 a discriminator for comparing the rewriting blocks with blocks executed by the processor to discriminate execution states of the blocks corresponding to the rewriting blocks; and

10 a rewriter for performing rewrite processing in which the corresponding blocks stored in the software storage are sequentially rewritten to the rewriting blocks in accordance with a discrimination result.

15 6. The software rewriting apparatus according to claim 5, wherein the discriminator comprises a table including configuration items having an item indicative of block numbers of the rewriting blocks and an item indicative of execution states of the blocks corresponding to the rewriting blocks, and  
20 the rewriter performs rewrite processing with reference to the table.

7. The software rewriting apparatus according to claim 6, wherein the table including a configuration item having an item indicative of rewriting states  
25 of the blocks corresponding to the rewriting blocks.

8. The software rewriting apparatus according to claim 5, further comprising a controller for surveying a degree of load of processing executed

by a CPU, and instructing the rewriter to carry out rewrite processing when the degree of load becomes low.

9. The software rewriting apparatus according to  
5 claim 8, wherein the controller surveys the degree of load of processing executed by the CPU in response to a rewrite request sent from the rewriter.

10. A communication terminal apparatus having a software rewriting apparatus, said software  
10 rewriting apparatus comprising:

a software storage for storing software having one block or a plurality of divided blocks;

a processor for expanding the blocks to be executed;

15 a block storage for temporarily storing rewriting blocks;

a discriminator for comparing the rewriting blocks with blocks executed by the processor to discriminate execution states of the blocks  
20 corresponding to the rewriting blocks; and

a rewriter for performing rewrite processing in which the corresponding blocks stored in the software storage are sequentially rewritten to the rewriting blocks in accordance with a  
25 discrimination result.

## ABSTRACT

A normal processing section 102 expands a program stored in a software storage 108 to an execution block storage 105 on a block unit basis to be executed, a rewrite block storage 106 temporarily stores rewriting blocks sent from a transfer section 109, a rewrite table 107 has an item indicative of execution states of the rewriting blocks. A rewrite processing section 103 discriminates unexecuted blocks in software by referring to the rewrite table 107, and rewrites unexecuted blocks of the program stored in the software storage 108 to the rewriting blocks stored in the rewrite block storage 106.

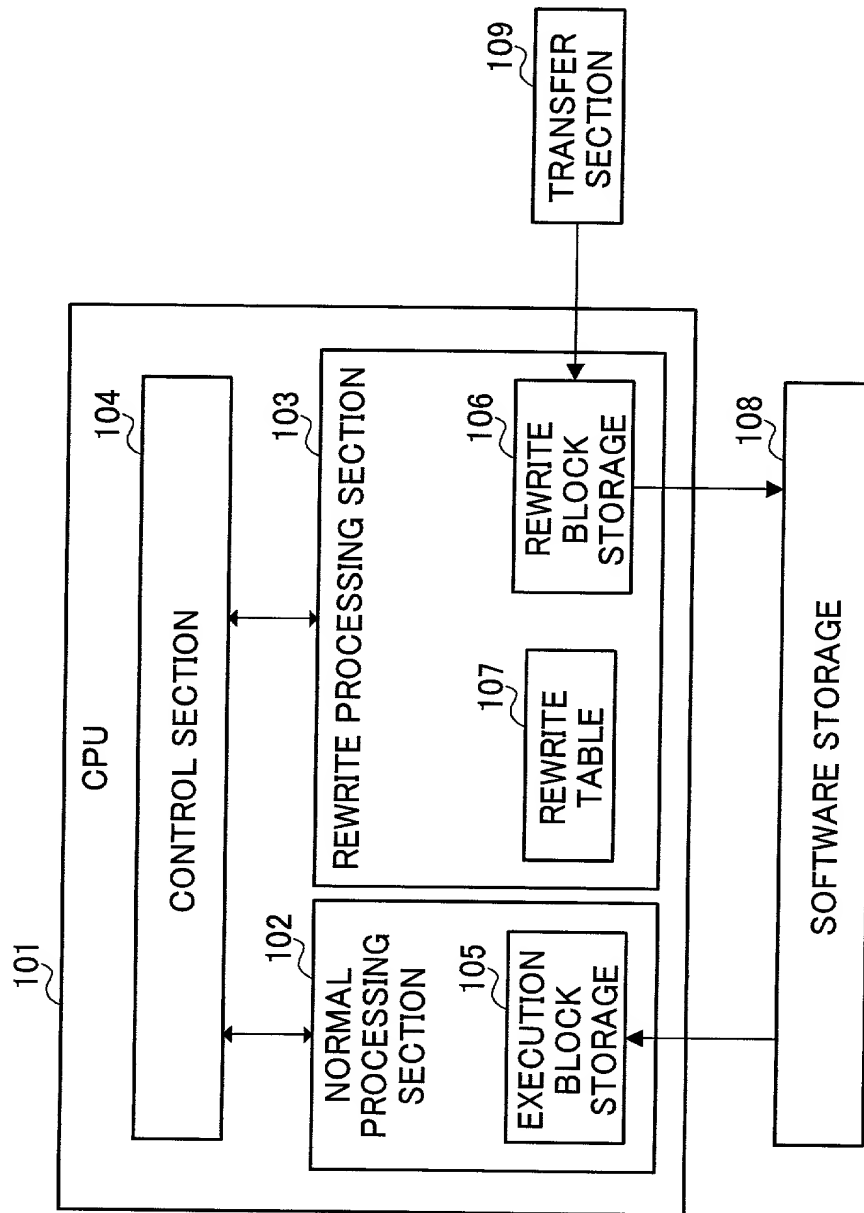


FIG.1

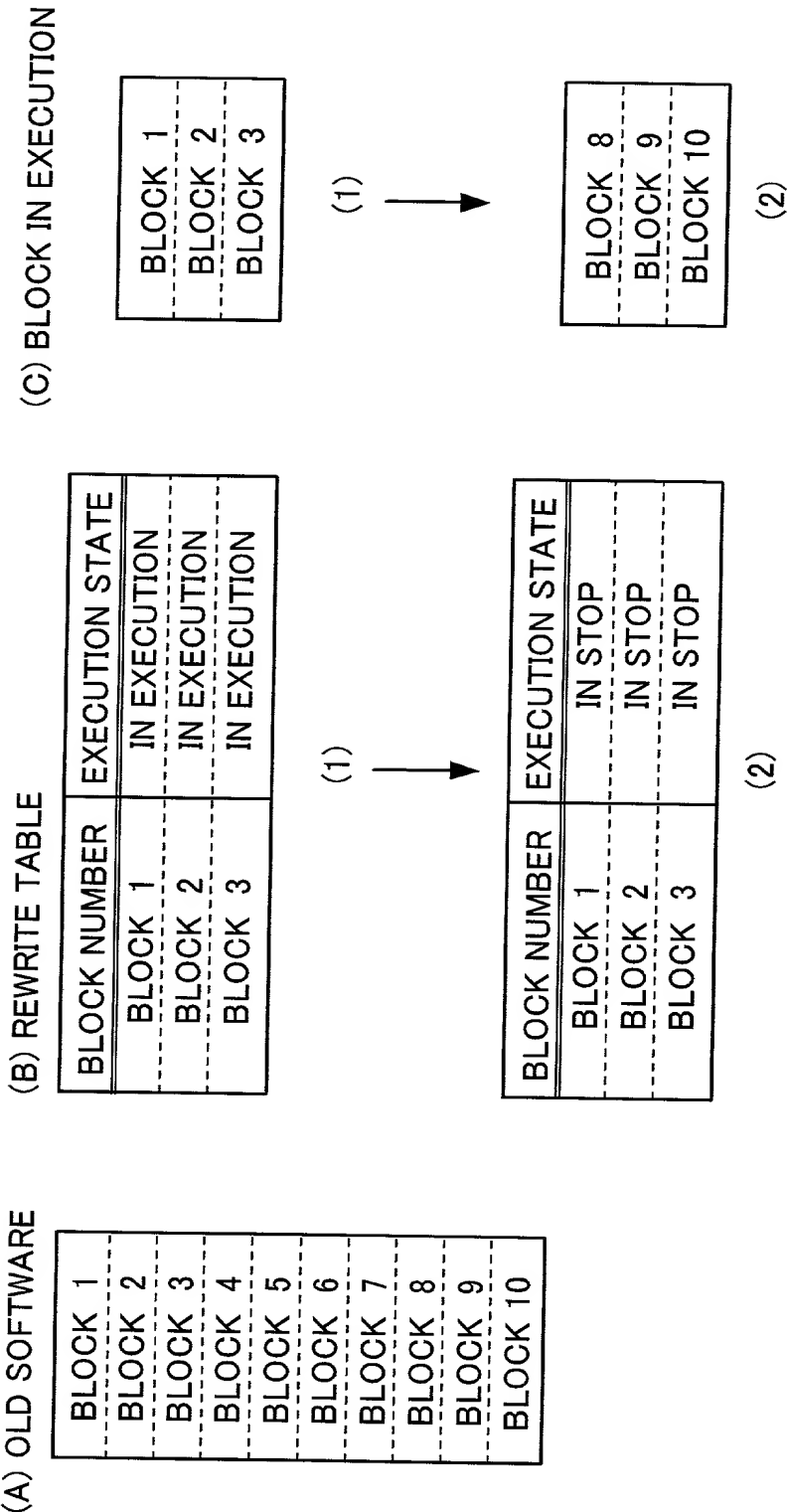


FIG.2

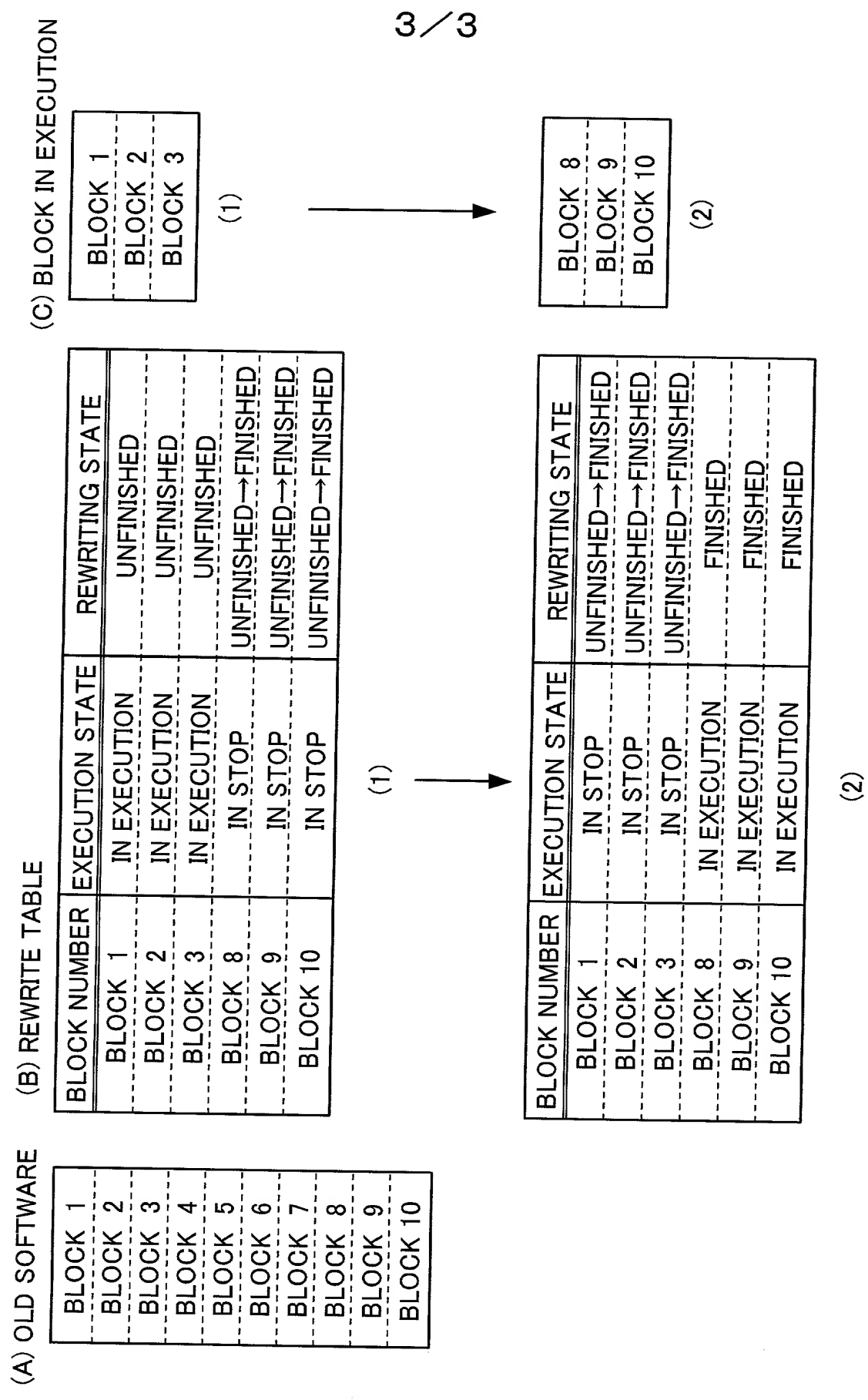


FIG.3

**APPLICATION FOR UNITED STATES PATENT**  
**Declaration for Patent Application**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on

the invention entitled: SOFTWARE REWRITING METHOD AND SOFTWARE REWRITING APPARATUS

the specification of which 2 (file no \_\_\_\_\_ )

(check at least one) 3 ☒ is attached hereto  
4 ☐ was filed on \_\_\_\_\_ as (5) U.S. Application Serial No. \_\_\_\_\_  
6 ☐ and was amended \_\_\_\_\_  
(if applicable)

Use this portion only if you are entering the U.S. National phase based on a PCT International Application designating the U.S.	7 <input checked="" type="checkbox"/>	was filed as PCT international application		
	8	Number	<u>PCT/JP00/05351</u>	
	9	on	<u>August 10, 2000</u>	
	10	and was amended under PCT Article(s) 19 and/or 34 on _____ (if applicable).		
	11	priority date claimed in PCT International Application	<u>JAPAN</u> (Country)	<u>JP11-231468</u> (Number)

I hereby declare that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended, by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application (s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date earlier than that of the application(s) on which priority is claimed.

Prior (Foreign) Application(s) any Priority Claims Under 35 U.S.C. 119 \_\_\_\_\_ Priority Claimed \_\_\_\_\_

12a	_____	_____	_____	<input type="checkbox"/>	<input type="checkbox"/>
	(Country)	(Number)	(Day/Month/Year Filed)	Yes	No
	_____	_____	_____	<input type="checkbox"/>	<input type="checkbox"/>
	(Country)	(Number)	(Day/Month/Year Filed)	Yes	No

Priority Claim(s) from U.S. Provisional Application(s) – I hereby claim the benefit under Title 35, United States Code, §119(e) of any United States provisional application(s) listed below:

12b	_____	_____	_____	_____
	Application No.	Day/Month/Year Filed	Application No.	Day/Month/Year Filed

Do not use this portion to identify a PCT application if the parent application is the U.S. National phase of the PCT application	I hereby claim the benefit under Title 35, United States Code, 120 of any United States application(s) or PCT international application(s) designating the United States of America that is/are listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in that/those prior application(s) in the manner provided by the first paragraph of Title 35, United States Code §112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, §1.56 which became available between filing date of the prior application and the national or PCT international filing date of this application.		
	13	_____	_____
	(U.S. Application Number)	(U.S. Filing Date)	Status (patented, pending, abandoned)

I hereby appoint the following attorneys of the firm of Stevens, Davis, Miller & Mosher, L.L.P. as my attorneys of record with full power of substitution and revocation to prosecute this application and to transact all business in the Patent and Trademark Office:

3 James E. Ledbetter, Reg. No. 28732; Thomas P. Pavelko, Reg. No. 31689; and Anthony P. Venturino, Reg. No. 31674.

**ALL CORRESPONDENCE IN CONNECTION WITH THIS APPLICATION SHOULD BE SENT TO**  
**STEVENS, DAVIS, MILLER & MOSHER, L.L.P., 1615 L Street, N.W., Suite 850, Washington, D.C. 20036,**  
**TELEPHONE (202) 408-5100, FACSIMILE (202) 408-5200.**

See page 2 for signature lines

# INSTRUCTIONS FOR COMPLETION OF THIS FORM

- line 1 Insert the same title as is used on the specification and in the assignment.
- line 2 Is optional but is provided so that you can use it to identify more readily an application prior to the time that the Patent Office application serial number is assigned. We suggest that the specification, drawings and declaration always bear a file number since it can help to get the papers together in case they become inadvertently separated. In instances where the specification is filed without a signed declaration form (under 37 CFR §1.53) a file number on a later-received separate form will assist us in associating it with the correct case.
- line 3 Check this box if the specification, claims and drawing (if any) are attached to this declaration form, e.g., when filing a new patent application.
- lines 4-5 Are only used in an instance where the application is already on file and the declaration from is being separately filed, e.g., when the application was originally filed without a signed declaration or where the Patent Office has required a new declaration because of a deficiency in the original declaration. In such an instance the Patent Office will require that lines 4 and 5 be completed with the filing date and application serial number already assigned.
- line 6 Is used in conjunction with line 5 but only when there have been one or more amendments to the specification or claims. Line 6 is also used when the Examiner requires a new declaration because claims inserted by amendment cover subject matter not originally claimed (37 CFR §1.67).
- lines 7-11 Are for PCT (Patent Cooperation Treaty) cases and are used only when you are entering the U.S. National phase (Chapter I or II) based upon a previously filed PCT International application designating the U.S.
- line 7 Check this box if this is a PCT National Phase application.
- line 8 Insert PCT International application number.
- line 9 Insert date of filing of PCT International application.
- lines 10-11 Insert the date of all amendments filed in the PCT International application. Such amendments are optional, so this line at times will not be used.
- line 12a Is used in the following instances:
- (i) If a single priority is being claimed from a foreign application you need to list only the first-filed application; you do not need to list other countries if all applications were filed within one year of the U.S. filing.
  - (ii) If multiple priorities are being claimed, from a plurality of applications filed in one or more countries, you must list the first filed application for each aspect of the invention. Example: if aspect A of the invention was disclosed in an application filed 11 months earlier in country X and aspect B was disclosed 9 months earlier in an application filed in country Y, then the applications in both countries X and Y must be identified. Only the first application for each aspect of the invention needs to be identified provided all applications on that aspect were filed within one year prior to the U.S. filing.
  - (iii) If a non-priority application is being filed you must list all applications in all countries where corresponding foreign applications were filed more than one year prior to the U.S. filing. This is so the Examiner can check to see if any of those applications were published or patented early enough to be prior art against the U.S. application.
  - (iv) If there are more than two applications to be listed we suggest that you type in on this form only "See attached Schedule A" and then list all of the previous applications on an attached sheet.
- line 12b Is used to claim priority under 35 USC §119(e) based on a provisional application filed within one year of the filing of the instant application. More than one provisional application may be identified provided neither was filed more than one year earlier.
- line 13 This block is used only in instances where there is a previously filed U.S. non-provisional application which was copending at the time the present application was (or is being) filed. That previous application could be a U.S. non-provisional application or the National Phase of a PCT allocation. In such a case the present application may be entitled to the priority of the previous application's U.S. filing date (and consequently the foreign priority thereof) provided the present application is identified as a continuing application (continuation, divisional or continuation-in-part) of the earlier (parent) application. If the foregoing is applicable, please fill in one line for each such prior application.
- line 14 Type the inventor's proper legal name in the order specified, e.g., "John B. JONES" or "J. Bob JONES" if the inventor so prefers. It is not acceptable to use only initials such as "J. B. JONES."
- line 15 The inventor's "signature" may be his (or her) usual manner of signing but it is preferable that the inventor simply write his (or her) name in his (or her) own cursive handwriting in the same order as on line 14, e.g., given name, middle initial and Family name.
- line 16 Insert the actual date of signature.
- line 17 Insert simply the city and state or country, e.g., "Paris, France", of the inventor's residence, not citizenship. No street address or postal code is required on this line.
- line 18 Insert the inventor's citizenship. The statement of citizenship (or subject of) is a statutory requirement (35 USC §115). Simply the name of the country of citizenship, e.g., "Japan" is sufficient.
- line 19 Insert the inventor's mailing address. The purpose of requiring the post office address is to enable the Patent Office to communicate directly with the inventor if desired, such as in the case of death of the U.S. attorney. It should be the address where the inventor customarily receives his (or her) mail and should include the postal code. If applicable it can be the inventor's business address or address at place of employment.

Applicants are reminded that the U.S. Patent and Trademark Office has very strict requirements as to proper execution of an application. The applicant should make sure that he reviews the declaration, prior to signing to make sure the declaration properly identifies the application and all relevant information; and should review the specification and claims (including drawings, if any) before signing the declaration. Failure to do so will require the filing of a supplemental declaration --- 37 CFR §1.67(c).

Any handwritten changes to the specification, claims or drawings must be in ink personally by all of the inventors prior to signing the declaration and the adjacent left margin must be initialed and dated by all of the inventors, e.g., "JBJ 6-9-91".

Please let us know if there are any questions regarding proper completion of this form. Thank you.

An assignment, a separate document requiring separate signature and dating may be enclosed. Please look for it and sign and date it in the same manner as in lines 15 and 16 above.

**STEVENSON, DAVIS, MILLER & MOSHER, L.L.P.**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful statements may jeopardize the validity of the application or any patent issuing thereon.

**PAGE 2 OF U.S.A. DECLARATION FORM**

14a	Typewritten Full Name of Sole or First Inventor	1-10 <u>Isao</u> <u>ISOGAI</u>		
		Given Name	Middle Name	Family Name
15a	Inventor's Signature	<u>Isao Isogai</u>		
16a	Date of Signature	<u>January</u>	<u>25</u>	<u>2001</u>
		Month	Day	Year
17a	Residence	<u>Yokosuka-shi</u>	<u>Kanagawa</u>	<u>JAPAN</u>
		City	State or Province	Country
18a	Citizenship	<u>JAPAN</u>		
19a	Post Office Address (Insert complete mailing address, including country)	<u>2-20-22-201, Tsukui, Yokosuka-shi,</u> <u>Kanagawa 239-0843 JAPAN</u>		
14b	Typewritten Full Name of Sole or First Inventor			
		Given Name	Middle Name	Family Name
15b	Inventor's Signature			
16b	Date of Signature			
		Month	Day	Year
17b	Residence			
		City	State or Province	Country
18b	Citizenship			
19b	Post Office Address (Insert complete mailing address, including country)			
14c	Typewritten Full Name of Sole or First Inventor			
		Given Name	Middle Name	Family Name
15c	Inventor's Signature			
16c	Date of Signature			
		Month	Day	Year
17c	Residence			
		City	State or Province	Country
18c	Citizenship			
19c	Post Office Address (Insert complete mailing address, including country)			
14d	Typewritten Full Name of Sole or First Inventor			
		Given Name	Middle Name	Family Name
15d	Inventor's Signature			
16d	Date of Signature			
		Month	Day	Year
17d	Residence			
		City	State or Province	Country
18d	Citizenship			
19d	Post Office Address (Insert complete mailing address, including country)			

\*Note to Inventor: Please sign name on line 15 exactly as it appears in line 14 and insert the actual date of signing on line 16. If there are more than four inventors, please add a copy of this page for identification and signatures for the additional inventors.